

日本国特許庁
JAPAN PATENT OFFICE

09.06.2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 6月10日
Date of Application:

出願番号 特願2003-165742
Application Number:
[ST. 10/C]: [JP 2003-165742]

出願人 ソニー・エリクソン・モバイルコミュニケーションズ株式会社
Applicant(s): 社

REC'D 29 JUL 2004

WIPO

PCT

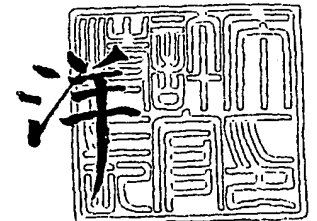
PRIORITY
DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

2004年 7月14日

特許庁長官
Commissioner,
Japan Patent Office

小川



【書類名】 特許願

【整理番号】 0300036004

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/46
H04N 5/44

【発明者】

【住所又は居所】 東京都港区港南 1 丁目 8 番 1 5 号 ソニー・エリクソン
・モバイルコミュニケーションズ株式会社内

【氏名】 大出 直樹

【特許出願人】

【識別番号】 501431073

【氏名又は名称】 ソニー・エリクソン・モバイルコミュニケーションズ株
式会社

【代理人】

【識別番号】 100107238

【弁理士】

【氏名又は名称】 米山 尚志

【手数料の表示】

【予納台帳番号】 111236

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 競合管理プログラム、競合管理プログラムが記憶された記憶媒体、競合管理方法及び電子機器

【特許請求の範囲】

【請求項1】 タスクの実行要求を受け付ける実行要求受付手段と、
起動中のタスクを起動中タスクリストに登録する起動中タスク登録手段と、
上記実行要求受付手段として機能して上記タスクの実行要求を受け付けた際に、
上記起動中タスクリストを参照して、タスク同士の競合の有無を検出する競合
検出手段と、

上記競合検出手段として機能することで上記タスク同士の競合を検出した際に、
該競合する各タスクの遷移すべき状態が記憶された競合条件テーブルを参照し
て、上記実行要求がなされたタスク、及び上記起動中タスクリストに登録されて
いるタスクの遷移させるべき状態をそれぞれ決定する遷移状態決定手段と、

上記実行要求がなされたタスク、及び上記起動中タスクリストに登録されてい
るタスクの状態を、上記遷移状態決定手段として機能することで決定した状態に
それぞれ遷移させる状態遷移手段として

コンピュータを機能させる競合管理プログラム。

【請求項2】 請求項1記載の競合管理プログラムであって、
上記起動中タスク登録手段としてコンピュータを機能させる際に、実行するタ
スクを上記起動中タスクリスト内の実行リストに登録し、実行待ちとするタスク
を上記起動中タスクリスト内の実行待ちリストに登録する起動中タスク登録手段
としてコンピュータを機能させること
を特徴とする競合管理プログラム。

【請求項3】 請求項2記載の競合管理プログラムであって、
上記起動中タスク登録手段としてコンピュータを機能させる際に、上記遷移状
態決定手段として機能することで決定された状態の優先度に基づいて、実行する
タスク及び実行待ちとするタスクの選別を行い、該実行するタスクを上記実行リ
ストに登録し、該実行待ちとするタスクを上記実行待ちリストに登録する起動中
タスク登録手段としてコンピュータを機能させること

を特徴とする競合管理プログラム。

【請求項4】 請求項1から請求項3のうち、いずれか一項記載の競合管理プログラムであって、

上記遷移状態決定手段としてコンピュータが機能することで、上記実行要求のなされたタスクの実行は不可であると決定された場合に、該タスクの実行要求をキャンセルする遷移状態決定手段としてコンピュータを機能させること

を特徴とする競合管理プログラム。

【請求項5】 タスクの実行要求を受け付ける実行要求受付手段と、

起動中のタスクを起動中タスクリストに登録する起動中タスク登録手段と、

上記実行要求受付手段として機能して上記タスクの実行要求を受け付けた際に、上記起動中タスクリストを参照して、タスク同士の競合の有無を検出する競合検出手段と、

上記競合検出手段として機能することで上記タスク同士の競合を検出した際に、該競合する各タスクの遷移すべき状態が記憶された競合条件テーブルを参照して、上記実行要求がなされたタスク、及び上記起動中タスクリストに登録されているタスクの遷移させるべき状態をそれぞれ決定する遷移状態決定手段と、

上記実行要求がなされたタスク、及び上記起動中タスクリストに登録されているタスクの状態を、上記遷移状態決定手段として機能することで決定した状態にそれぞれ遷移させる状態遷移手段として

コンピュータを機能させる競合管理プログラムが記憶されたコンピュータ読み取り可能な記憶媒体。

【請求項6】 請求項5記載の記憶媒体であって、

上記起動中タスク登録手段としてコンピュータを機能させる際に、実行するタスクを上記起動中タスクリスト内の実行リストに登録し、実行待ちとするタスクを上記起動中タスクリスト内の実行待ちリストに登録する起動中タスク登録手段としてコンピュータを機能させること

を特徴とする競合管理プログラムが記憶されたコンピュータ読み取り可能な記憶媒体。

【請求項7】 請求項6記載の記憶媒体であって、

上記起動中タスク登録手段としてコンピュータを機能させる際に、上記遷移状態決定手段として機能することで決定された状態の優先度に基づいて、実行するタスク及び実行待ちとするタスクの選別を行い、該実行するタスクを上記実行リストに登録し、該実行待ちとするタスクを上記実行待ちリストに登録する起動中タスク登録手段としてコンピュータを機能させること

を特徴とする競合管理プログラムが記憶されたコンピュータ読み取り可能な記憶媒体。

【請求項 8】 請求項 5 から請求項 7 のうち、いずれか一項記載の記憶媒体であって、

上記遷移状態決定手段としてコンピュータを機能させることで、上記実行要求のなされたタスクの実行は不可であると決定された場合に、該タスクの実行要求をキャンセルする遷移状態決定手段としてコンピュータを機能させること

を特徴とする競合管理プログラムが記憶されたコンピュータ読み取り可能な記憶媒体。

【請求項 9】 実行要求受付手段がタスクの実行要求を受け付けるステップと、

起動中タスク登録手段が、起動中のタスクを起動中タスクリストに登録するステップと、

上記実行要求受付手段が上記タスクの実行要求を受け付けた際に、競合検出手段が、上記起動中タスクリストを参照して、タスク同士の競合の有無を検出するステップと、

上記競合検出手段が上記タスク同士の競合を検出した際に、遷移状態決定手段が、該競合する各タスクの遷移すべき状態が記憶された競合条件テーブルを参照して、上記実行要求がなされたタスク、及び上記起動中タスクリストに登録されているタスクの遷移させるべき状態をそれぞれ決定するステップと、

上記実行要求がなされたタスク、及び上記起動中タスクリストに登録されているタスクの状態を、状態遷移手段が、上記遷移状態決定手段で決定された状態にそれぞれ遷移させるステップと

を有する競合管理方法。

【請求項 10】 請求項 9 記載の競合管理方法であって、

上記起動中のタスクを起動中タスクリストに登録するステップでは、起動中タスク登録手段が、実行するタスクを上記起動中タスクリスト内の実行リストに登録し、実行待ちとするタスクを上記起動中タスクリスト内の実行待ちリストに登録すること

を特徴とする競合管理方法。

【請求項 11】 請求項 10 記載の競合管理方法であって、

上記起動中のタスクを起動中タスクリストに登録するステップでは、起動中タスク登録手段が、上記遷移状態決定手段で決定された状態の優先度に基づいて、実行するタスク及び実行待ちとするタスクの選別を行い、該実行するタスクを上記実行リストに登録し、該実行待ちとするタスクを上記実行待ちリストに登録すること

を特徴とする競合管理方法。

【請求項 12】 請求項 9 から請求項 11 のうち、いずれか一項記載の競合管理方法であって、

上記遷移状態決定手段で決定された状態に遷移させるステップでは、上記遷移状態決定手段が、上記実行要求がなされたタスクの実行は不可であると決定した場合に、該タスクの実行要求をキャンセルすること

を特徴とする競合管理方法。

【請求項 13】 タスクの実行要求を受け付ける実行要求受付手段と、

起動中のタスクを起動中タスクリストに登録する起動中タスク登録手段と、

上記実行要求受付手段で上記タスクの実行要求を受け付けた際に、上記起動中タスクリストを参照して、タスク同士の競合の有無を検出する競合検出手段と、

上記競合検出手段で上記タスク同士の競合が検出された際に、該競合する各タスクの遷移すべき状態が記憶された競合条件テーブルを参照して、上記実行要求がなされたタスク、及び上記起動中タスクリストに登録されているタスクの遷移させるべき状態をそれぞれ決定する遷移状態決定手段と、

上記実行要求がなされたタスク、及び上記起動中タスクリストに登録されているタスクの状態を、上記遷移状態決定手段で決定された状態にそれぞれ遷移させ

る状態遷移手段と

を有する電子機器。

【請求項 14】 請求項 13 記載の電子機器であって、

上記起動中タスク登録手段は、実行するタスクを上記起動中タスクリスト内の実行リストに登録し、実行待ちとするタスクを上記起動中タスクリスト内の実行待ちリストに登録すること

を特徴とする電子機器。

【請求項 15】 請求項 14 記載の電子機器であって、

上記起動中タスク登録手段は、上記遷移状態決定手段で決定された状態の優先度に基づいて、実行するタスク及び実行待ちとするタスクの選別を行い、該実行するタスクを上記実行リストに登録し、該実行待ちとするタスクを上記実行待ちリストに登録すること

を特徴とする電子機器。

【請求項 16】 請求項 13 から請求項 15 のうち、いずれか一項記載の電子機器であって、

上記遷移状態決定手段は、上記実行要求がなされたタスクの実行は不可であると決定した場合に、該タスクの実行要求をキャンセルすること

を特徴とする電子機器。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、例えば携帯電話機、PHS 電話機 (PHS: Personal Handyphone System), PDA 装置 (PDA: Personal Digital (Data) Assistant), パーソナルコンピュータ装置等の端末装置に設けて好適な競合管理プログラム; 競合管理プログラムが記憶された記憶媒体、競合管理方法及び電子機器に関する。

【0002】

【従来の技術】

今日における携帯電話機は、シングルタスクのかたちでアプリケーションの実行管理を行うのが一般的である。

【0 0 0 3】

例えば、一方のアプリケーションの実行中に、他方のアプリケーションの実行指示があった場合、これら各アプリケーションの優先度をそれぞれ検出し、優先度の低いアプリケーションを強制的に終了させ、代わりに優先度の高いアプリケーションを起動することで、該アプリケーションの競合管理を行うようになっている。

【0 0 0 4】

具体的には、例えば電子メールの作成中に着呼があった場合、電子メールを作成するアプリケーションよりも、着呼を処理するアプリケーションの方が優先度が高いため、電子メールを作成するアプリケーションを強制的に終了させ、着呼を処理するアプリケーションを起動して着呼の処理を行う。これにより、電子メールの作成は中止され、着呼が優先して処理されることとなる。

【0 0 0 5】

これに対して、携帯電話機よりも情報処理能力（パフォーマンス）の高いパーソナルコンピュータ装置の場合、マルチタスクのかたちでアプリケーションの実行管理を行うのが一般的である。

【0 0 0 6】

このパーソナルコンピュータ装置においては、アプリケーションの競合が生じた場合、以下のいずれかの方法で競合管理を行うようになっている。

【0 0 0 7】

まず、第1の競合管理方法として、オペレーティングシステム（OS）が、実行中のアプリケーションの現在の状態にかかわらず、該実行中のアプリケーションの優先度を変更し（優先度を下げ）、後から実行が指定されたアプリケーションを優先して実行する競合管理を行う方法である。

【0 0 0 8】

具体的には、例えば表計算ソフトウェアの実行中に、ユーザから文書作成ソフトウェアの立ち上げが指示された場合、OSは、表計算ソフトウェアのアプリケーションを待機状態とし、文書作成ソフトウェアのアプリケーションを優先して起動する。これにより、表計算ソフトウェアによる表作成を中断した状態で、文

書作成ソフトウェアによる文書作成が可能となる。

【0009】

次に、第2の競合管理方法として、アプリケーションのプログラミングの段階で競合する他のアプリケーションを予測し、競合が生じた場合、アプリケーション同士で通信を行い、一方のアプリケーションが他方のアプリケーションの状態に応じて優先度を変更するように、該アプリケーションをプログラミングしておくことで競合管理を行う方法である。

【0010】

この第2の競合管理方法においては、競合したアプリケーション同士の通信で優先順位が決まるため、各アプリケーションが自己的に競合管理を行うこととなる。

【0011】

なお、特表2003-504753号の公表特許公報に、このようなアプリケーション管理に関する先行技術が開示されている。

【0012】

【特許文献】

特表2003-504753号公報

【0013】

【発明が解決しようとする課題】

ここで、上述のように、現行における携帯電話機は、シングルタスクのかたちでアプリケーションの競合管理を行っているのであるが、将来的にマルチタスクのかたちでアプリケーションの実行管理を行うことが予測される。

【0014】

しかし、前述の第1の管理方法或いは第2の管理方法で競合管理を行うようにすると以下の問題を生ずる。

【0015】

すなわち、上記第1の競合管理方法で競合管理を行うと、OSがアプリケーションの現在の状態を認識することなく後先のかたちでアプリケーションが実行されることとなるため、後から実行指示のなされたアプリケーションの実行が終了

し、待機状態となっていたアプリケーション（＝先に実行されていたアプリケーション）の実行が再開される際に、該待機状態となっていたアプリケーションが正常に動作しないことが懸念される（待機状態とされたアプリケーションの動作保証が無い。）。

【0016】

また、上記第2の競合管理方法で競合管理を行う場合には、予め競合するアプリケーションを全て予測してアプリケーションのプログラミングを行う必要がある。このような予測は、アプリケーション数が増え、また、システムが複雑化する程困難となる。

【0017】

また、上記第2の競合管理方法で競合管理を行うこととすると、アプリケーションに仕様変更が生じた場合には、他の全てのアプリケーションも、この仕様変更に対応した仕様変更が必要となり、多数のアプリケーションの再プログラミング作業や再インストール作業（アプリケーションの設計及び実装の変更作業）等の大変面倒な作業が必要となることが懸念される。

【0018】

マルチタスクを可能とする従来の競合管理方法は、このようなことを考慮する必要があったため、ソフトウェアの開発を困難なものとする問題があった。

【0019】

本発明は、上述の課題に鑑みてなされたものであり、待機状態としたアプリケーションの動作を保証すること、競合するアプリケーションを考慮することなくアプリケーションのプログラミング（設計）を可能とすること、仕様変更が生じた場合でも他のアプリケーションの仕様変更を必要としないことを通じてソフトウェアの開発を容易化可能な競合管理プログラム、競合管理プログラムが記憶された記憶媒体、競合管理方法及び電子機器の提供を目的とする。

【0020】

【課題を解決するための手段】

本発明は、実行要求受付手段がタスクの実行要求を受け付けた際に、競合検出手段が、起動中のタスクが登録されている起動中タスクリストを参照して、タス

ク同士の競合の有無を検出する。

【0021】

上記競合検出手段で上記タスク同士の競合が検出されると、遷移状態決定手段が、該競合する各タスクの遷移すべき状態が記憶された競合条件テーブルを参照して、上記実行要求がなされたタスク、及び上記起動中タスクリストに登録されているタスクの遷移させるべき状態をそれぞれ決定する。

【0022】

そして、上記実行要求がなされたタスク、及び上記起動中タスクリストに登録されているタスクの状態を、状態遷移手段が、上記遷移状態決定手段で決定された状態にそれぞれ遷移させる。

【0023】

なお、ここでいう「タスク」の概念は、アプリケーションの他、コンピュータが情報処理を行う際に実行する全てのタスクを含む概念である。

【0024】

また、「起動中のタスク」の概念は、現在実行されているタスク、及び起動されてはいるが、現在、実行待ちとされているタスクの両方を含む概念である。

【0025】

【発明の実施の形態】

本発明は一例として、携帯電話機に適用することができる。この本発明の実施の形態となる携帯電話機は、いわゆるホームメモリ機能、位置登録機能、追跡交換機能（自動追跡ルーチン）及びハンドオーバ機能等を備えた移動体通信システムで用いられるものである。

【0026】

この移動体通信システムのサービスエリアには、複数の基地局が分散配置されており、これらの基地局によりそれぞれセルと呼ばれる無線通信エリアが形成されている。各基地局は、自局の無線通信エリア内に位置する各携帯電話機に対して無線チャンネルを割り当て、この無線チャンネルを介して各携帯電話機と通信を行い、該各携帯電話機とそれぞれ同期をとるようになっている。

【0027】

また、この移動体通信端末システムは、通信中に携帯電話機がセル間を移動した場合、移動前のセルの基地局から移動後のセルの基地局に、通信に用いる基地局を切り替える、いわゆるハンドオーバを行って移動しながらの通信を可能としている。

【0028】

図1は、CDMA方式(CDMA:Code Division Multiple Access)の移動体通信システムで用いられる当該第1の実施の形態の携帯電話機の全体的なブロック図である。

【0029】

[受信動作]

この図1において、基地局から送信された無線周波信号は、アンテナ1で受信され、アンテナ共用器2(DUP)を介して受信回路3(RX)に供給される。RX3は、上記無線周波信号と、周波数シンセサイザ4(SYN)から供給された受信局部発振信号とをミキシング処理し、該無線周波信号を中間周波信号に周波数変換処理する。

【0030】


なお、SYN4から出力される受信局部発振信号の周波数は、制御部12(CPU)からの制御信号(SYC)により設定される。

【0031】

上記受信中間周波信号は、CDMA信号処理部6において、直交復調処理が施されたのち、受信チャネルに割り当てられた拡散符号(PN符号)により逆拡散処理され、これによりデータレートに応じた所定のフォーマットの復調データに変換される。そして、この変換された復調データはデータ符号処理部7に供給され、また上記受信データのうちデータレートを示す受信データレートはCPU12に供給される。

【0032】

データ符号処理部7は、上記CDMA信号処理部6から供給された復調データに対し、CPU12から通知される受信データレートに応じた伸長処理を施した後、ビタビ復号等を用いた復号処理及び誤り訂正復号処理を施し、ベースバンド



の受信データを形成する。

【0033】

P C M符号処理部 8 は、音声信号通信の種別（音声通信、データ通信）に応じて異なる信号処理を行なうように、C P U 1 2 により制御される。

【0034】

具体的には、音声通信時には、P C M符号処理部 8 は、データ符号処理部 7 から供給された受信データを P C M復号処理してアナログ受話信号を形成する。このアナログ受話信号は、受話増幅器 9 にて増幅された後、スピーカ 1 0 を介して拡声出力される。

【0035】

また、データ通信時には、P C M符号処理部 8 は、データ符号処理部 7 からの受信データを C P U 1 2 へ供給する。C P U 1 2 は、この受信データをメモリ 1 3 に記憶制御する。また必要に応じて、上記受信データを、外部インタフェースを介して、携帯情報端末装置（PDA: Personal Digital Assistance）やノート型のパーソナルコンピュータ装置へ出力する。

【0036】**[送信動作]**

次に、音声通信時における話者の送話音声は、マイクロホン 1 1 により集音され、送話増幅器 1 8 で適正レベルに増幅される。そして、P C M符号処理部 8 で P C M符号化処理が施された後、送信データとしてデータ符号処理部 7 に供給される。

【0037】

なお、上記外部インタフェースを介して、例えば上記 P D A、ノート型のパーソナルコンピュータ装置、カメラ装置等の外部機器からデータや画像データ等が供給された場合、これらのデータは、C P U 1 2 に供給され、この C P U 1 2 から P C M符号処理部 8 を介してデータ符号処理部 7 に供給される。

【0038】

データ符号処理部 7 は、音声通信時には、P C M符号処理部 8 から供給された送信音声データに基づいて入力音声のエネルギー量を検出し、この検出結果に基づ

いてデータレートを決定する。そして、上記送信データをこの決定したデータレートに応じたフォーマットのバースト信号に圧縮処理し、さらに誤り訂正符号化処理を施したのちCDMA信号処理部6に供給する。

【0039】

また、データ通信時には、データ符号処理部7は、PCM符号処理部8から供給された送信データを、予め設定されたデータレートに応じたフォーマットのバースト信号に圧縮処理し、さらに誤り訂正符号化処理を施してCDMA信号処理部6に供給する。

【0040】

なお、音声通信時及びデータ通信時のいずれの場合も、上記データレートである送信データレートを示す情報は、CPU12に供給される。

【0041】

CDMA信号処理部6は、データ符号処理部7で圧縮されたバースト信号に対して、送信チャネルに割り当てられたPN符号を用いた拡散処理を施す。そしてこの拡散符号化された送信信号に対して直交変調処理を施し、この直交変調信号を送信回路5(TX)に供給する。

【0042】


送信回路5は、この直交変調信号を、SYN4から供給される送信局部発振信号とミキシング処理することで無線周波信号に変換する。そして、送信回路5は、CPU12により設定される送信データレートに基づいて、上記無線周波信号の有効部分だけを高周波増幅処理し、送信無線周波信号として、DUP2を介してアンテナ1に供給する。これにより、この送信無線周波信号が、アンテナ1から基地局に対してバースト送信されることとなる。

【0043】

なお、操作部14(KEY)には、ダイヤルキー、発信キー、電源キー、終了キー、音量調節キー、モード指定キー等のキー群が設けられている。

【0044】

また表示部15(LCD)には、通話相手端末の電話番号や装置の動作状態、さらには受信データ等を表示するための液晶表示部の他、発信／着信動作や充電



池 16 (バッテリー) の充電動作等を示す発光部 (LED等) が設けられている。

【0045】

また、電源回路 17 は、バッテリー 16 に蓄電されている電力に基づいて、所定の動作電源電圧 V_{cc} を形成し、これを当該携帯電話機の各回路部に供給するようになっている。

【0046】

[主要部の構成]

次に、図 2 に当該実施の形態の携帯電話機の主要部の構成を示す。この図 2 からわかるように、当該実施の形態の携帯電話機は、上記 CPU 12 により動作制御されるハードウェア群 21 と、このハードウェア群 21 の各ハードウェアに対応する複数のドライバからなるドライバ群 22 と、オペレーションシステム 23 (OS) と、複数のアプリケーションからなるアプリケーション群 24 と、アプリケーションの競合管理を行うアプリケーションマネージャ 25 と、各アプリケーションの動作状態毎に、競合したアプリケーションの起動の是非等を示す情報が記憶された競合条件テーブル 26 とを有している。

【0047】

ハードウェア群 21 は、上記操作部 14 及び上記表示部 15 と、上記 DUP 2, RX 3, CDMA 信号処理部 6, データ符号処理部 7 等の通信システム 30 と、携帯カメラ装置 31 と、発光部 (LED) や音声デバイス等の他のハードウェア 32 と、上記メモリ 13 等のメモリ/ファイルシステム 33 とを有している。

【0048】

ドライバ群 22 は、上記表示部 15 を駆動制御する LCD ドライバ 34 (LCD Task/Driver) と、上記操作部 14 を駆動制御するキードライバ 35 (KEY Driver) と、上記通信システムを駆動制御する通信ドライバ 36 と、上記携帯カメラ装置 31 を駆動制御するカメラドライバ 37 (Camera Driver) と、上記他のハードウェアを駆動制御する他のハードウェアのドライバ 38 と、上記メモリ/ファイルシステム 33 を駆動制御するメモリ/ファイルシステムドライバ 39 とを有している。

【0049】

アプリケーション群 24 は、各アプリケーションが共用するデータ群であるライブラリ 40 (Libraries) と、ウインドウマネージャ 41 (Window Manager) と、各アプリケーションの基本データ群となるアプリケーションフレームワーク 42 (Application Framework) と、例えば電話アプリケーション (Tel App) や時計アプリケーション (Clock) , スクリーンセーバー (Screen Saver) , アプリケーションランチャー (App Launcher) 等の各アプリケーション 43 とを有している。

【0050】

なお、この図 2 における「App1」, 「App2」は、ユーザ定義の任意のアプリケーションを示している。

【0051】

また、この図 2 において、CPU 12, 操作部 14, 表示部 15, 通信システム 30, 携帯カメラ装置 31, 他のハードウェア 32 及びメモリ/ファイルシステム 33 がそれぞれ「ハードウェア」であるのに対し、OS 23 ~ 競合条件テーブル 26 は、それぞれ「ソフトウェア」となっており、例えば図 1 に示すメモリ 13 に格納されている。

【0052】

(各アプリケーションの機能構成)

次に、図 3 に上記アプリケーション群 24 の各アプリケーション 43 の機能ブロック図を示す。この図 3 に示すように、アプリケーション 43 は、当該アプリケーションのインターフェイスとなるコミュニケーションブロック 50 と、上記アプリケーションマネージャ 25 との間で、上記コミュニケーションブロック 50 を介して通信を行うシステム制御ブロック 51 と、例えば電話通信や携帯メールの作成等の、そのアプリケーションの動作を行うブロックであるユーザ制御ブロック 52 とを有している。

【0053】

後に説明するが、このアプリケーションは、システム制御ブロック 51 がアプリケーションマネージャ 25 との間で通信を行うことで、該アプリケーションマネージャ 25 により動作状態 (起動, 起動待機等) が制御されるようになっている。

る。

【0054】

(アプリケーションマネージャの機能構成)

次に、図4にアプリケーションマネージャ25の機能ブロック図を示す。この図4に示すように、アプリケーションマネージャ25は、当該アプリケーションマネージャ25のインターフェイスとなるコミュニケーションブロック60と、実行中のアプリケーションを示すデータが格納される実行リスト61及び実行待ちとなっているアプリケーションを示すデータが格納される実行待ちリスト62を備えたアプリケーションリスト63とを有している。

【0055】

また、このアプリケーションマネージャ25は、上記アプリケーションリストに格納されている各リスト61, 62に基づいてアプリケーションの競合管理を行うアプリケーション管理ブロック64と、このアプリケーション管理ブロック64の競合管理に対応して各アプリケーションの起動, 終了, 状態遷移等を制御するアプリケーション制御ブロック65とを有している。

【0056】

(アプリケーションマネージャの動作概要)

このようなアプリケーションマネージャ25は、各アプリケーションが競合した場合に、各アプリケーションの優先順位や現在の状態等に応じて、実行するアプリケーション、起動するアプリケーション、起動待ちとするアプリケーション、実行待機とするアプリケーションをそれぞれ決定し、それぞれ対応する上記リスト61, 62に登録する。

【0057】

そして、登録したアプリケーションの状態や新たなアプリケーションの起動要求等の状況の変化があった場合に、その現状に対応して上記各リスト61, 62を再構築し、この各リスト61, 62を参照しながら各アプリケーションの競合管理を行う。

【0058】

(アプリケーションの状態の遷移)

すなわち、もう少し詳しく説明するために、図5に、各アプリケーションの各状態を模式的に示す。この図5からわかるように、各アプリケーションの状態としては、例えば当該携帯電話機内に記憶されていない状態（或いはアプリケーションが記憶されたメモリが当該携帯電話に設けられていない状態）である「非存在状態（Non-Existent）」と、アプリケーション自体は記憶されているのであるが、実行が指定されていない状態である「不活性状態（Dormant）」とがある。

【0059】

また、各アプリケーションの状態としては、そのアプリケーションが現在、実行されている状態である「実行状態」、及びそのアプリケーションが実行待ちとなっている状態である「実行待ち状態」とがある。

【0060】

当該携帯電話機に設けられていないアプリケーションは、例えば当該携帯電話機にインストールされ、或いはそのアプリケーションが記憶されたメモリが当該携帯電話機に設けられることで、図5中「生成」として示すように、「不活性状態（当該携帯電話機内に存在しているのであるが、実行或いは実行の指示が無く休眠している状態）」となる。

【0061】

この「不活性状態」のアプリケーションは、例えばユーザや他のアプリケーションからの実行指示があった場合、図5中「起動」或いは「待機起動」として示すように、上記「実行状態」或いは「実行待ち状態」のいずれかの状態に遷移する。

【0062】

アプリケーションマネージャ25は、この「不活性状態」のアプリケーションを「実行状態」に遷移させた場合は、上記実行リスト61に、その実行状態としたアプリケーションを登録する。

【0063】

また、アプリケーションマネージャ25は、この「不活性状態」のアプリケーションを「実行待ち状態」に遷移させた場合は、上記実行待ちリスト62に、その実行待ち状態としたアプリケーションを登録する。

【0064】

次に、実行状態とされたアプリケーションは、その動作状況等に応じてその状態が遷移する。

【0065】

一例ではあるが、上記アプリケーションが電話アプリケーションの場合、通話先とのコネクションを確立するための処理を行う接続状態、コネクションが確立された後に通話中の音声の処理等を行う通話状態、通話中に該通話の保留が指定された場合に、通話の保留処理を行う保留状態、通話の終了が指定された場合に、確立されているコネクションを切断処理する切断状態等のように、通話の開始から終了までの間、アプリケーションの状態が変化する。

【0066】

図5中の実行状態における第1～第4の状態及び相互の矢印は、このような実行中のアプリケーションの状態の遷移を示している。

【0067】

上記アプリケーションマネージャ25は、「アプリケーションの起動要求を受け付けた際」、「実行中のアプリケーションの終了を検出した際」、「実行中のアプリケーションの状態遷移を検出した際」に、上記実行リスト61及び上記実行待ちリスト62に登録されているアプリケーションを参照し、各アプリケーションの「優先度」及び「状態」に応じて、例えば実行中のアプリケーションを待機状態とし、代わりに実行待ち状態のアプリケーションや新たに起動要求のなされたアプリケーションを実行状態し、或いは新たに起動要求のなされたアプリケーションは起動せずに起動待機状態とする等のように競合管理を行う。

【0068】

なお、実行が終了したアプリケーションは、図5に「終了」として示すように前述の「不活性状態」に遷移する。また、「不活性状態」のアプリケーションは、削除されると、前述のように携帯電話機内に存在しない状態である「非存在状態」となる。

【0069】

[アプリケーションマネージャの競合管理動作]

次に、このようなアプリケーションマネージャ25の競合管理動作を、図6のフローチャートを用いて説明する。

【0070】

この図6のフローチャートは、アプリケーションマネージャ25が、「アプリケーションの起動要求を受け付けた際」、「実行中のアプリケーションの終了を検出した際」、「実行中のアプリケーションの状態の遷移を検出した際」にスタート（条件判定開始）となり、ステップS1から各ステップを実行する。

【0071】

まず、ステップS1では、アプリケーションマネージャ25が、「アプリケーションの起動要求」、「実行中のアプリケーションの終了」、「実行中のアプリケーションの状態の遷移」のいずれかを検出したため、図4に示すアプリケーションマネージャ25のアプリケーション管理ブロック64がアプリケーションリスト63を参照する。

【0072】

次に、ステップS2では、アプリケーション管理ブロック64が、上記アプリケーションリスト63の実行リスト61に登録されているアプリケーションの有無を検出することで、現在、実行中のアプリケーションが存在するか否かを判別する。

【0073】

アプリケーション管理ブロック64は、上記実行リスト61を参照し、現在、実行中のアプリケーションが存在しなかった場合（＝実行リスト61にアプリケーションが登録されていなかった場合）、ステップS3において、起動要求のなされたアプリケーションを実行リスト61に登録し、或いは実行待ちリスト62に登録されている実行待ち状態のアプリケーションの中から実行状態に遷移させるアプリケーションを選択し、このアプリケーションを実行リスト61に登録する。

【0074】

このようにアプリケーション管理ブロック 64 により、実行リスト 61 に対して上記アプリケーションの登録が行われると、アプリケーション制御ブロック 65 は、ステップ S10 において、上記実行リスト 61 に登録されたアプリケーションを実行状態に遷移させる。

【0075】

これにより、起動要求のなされたアプリケーション、或いは実行待ちとなっていたアプリケーションが実行されることとなる。

【0076】

次に、上記ステップ S2 において、実行リスト 61 に一つ以上のアプリケーションの登録がされていた場合、アプリケーション管理ブロック 64 は、図 2 に示す競合条件テーブル 26 を参照する「競合チェック」を行う。

【0077】

(競合条件テーブル)

この「競合チェック」に用いられる競合条件テーブル 26 は、実行中のアプリケーションの一つ一つの状態を、競合する他のアプリケーションの一つ一つの状態にそれぞれ対応させたテーブルとなっており、この実行中のアプリケーションの状態と競合する他のアプリケーションの状態との競合関係において、実行中のアプリケーション及び競合する他のアプリケーションが遷移すべき状態を示したテーブルとなっている。

【0078】

従って、当該実施の形態の携帯電話機に設けられているアプリケーションの総数を「N」、各アプリケーションが有する状態数を「M」とした場合、当該実施の形態の携帯電話機に設けられる競合条件テーブルの総数は、「競合条件テーブルの総数 = $N \times M$ 」となる。

【0079】

具体的には、例えば図 7 は、現在実行中のアプリケーションが電話アプリケーションである場合に参照される競合条件テーブル 26 の一例を示している。

【0080】

この図 7 中、第 1 ～ 第 N の状態 (N は自然数) は、前述の接続状態、通話状態



、保留状態・・・の各状態を示している。また、図7中、「電話」、「待ち受け」・・・「メモ帳」は、状態が参照される各競合アプリケーションを示している。

【0081】

また、この図7に示す「起動不可」、「割込側待機」、「動作側待機」、「動作側終了」の各状態は以下のことを意味している。

【0082】

「起動不可」→起動要求のなされたアプリケーションは起動できないため、該起動要求をキャンセルする。

【0083】

「割込側待機」→起動要求のなされたアプリケーションを実行待ち状態とし、現在実行しているアプリケーションを継続して実行する。

【0084】

「動作側待機」→現在実行中のアプリケーションを終了することなく実行待ち状態とすると共に、起動要求のなされたアプリケーションの実行を開始する。

【0085】

「動作側終了」→現在実行中のアプリケーションを終了して、起動要求のなされたアプリケーションの実行を開始する。

【0086】

これら各状態は、

「起動不可」>「割込側待機」>「動作側待機」>「動作側終了」の順に優先度が割り当てられている。

【0087】

アプリケーション管理ブロック64は、実行中のアプリケーションの現在の状態、及び競合したアプリケーションの現在の状態に基づいて上記競合条件テーブルを参照して、実行中のアプリケーション及び競合したアプリケーションの遷移させるべき状態（上記起動不可、割込側待機、動作側待機、動作側終了等）をそれぞれ検出する。

【0088】



そして、アプリケーション制御ブロック 65 が、このアプリケーション管理ブロック 64 により検出された上記実行中のアプリケーションの遷移させるべき状態及び競合したアプリケーションの遷移させるべき状態の上記優先度に基づいて、該実行中のアプリケーションの状態を上記検出された遷移させるべき状態に遷移させ、また、該競合したアプリケーションの状態を上記検出された遷移させるべき状態に遷移させる。

【0089】

このような流れを、図 6 のフローチャートを用いて説明すると、アプリケーション管理ブロック 64 は、ステップ S5 において、起動要求のなされたアプリケーション（割込側アプリケーション）の起動は不可であるか否かを、上記競合チェックにより判別する。

【0090】

上記競合チェックにより、割込側アプリケーションの起動は不可であると判別した場合、割込側アプリケーションを起動することはできないため、アプリケーション管理ブロック 64 は、ステップ S10 において上記起動要求のなされたアプリケーションの該起動要求をキャンセルする。

【0091】


これに対して、上記競合チェックにより、割込側アプリケーションの起動は可能であると判別した場合、アプリケーション管理ブロック 64 は、この割込側アプリケーションの判別結果（アプリケーション ID、アプリケーション名、タスク ID、状態等）を一旦「仮リスト」に登録する。

【0092】

なお、この「仮リスト」に登録された上記判別結果は、アプリケーションリスト 63 に登録されている全てのアプリケーションに対して上記競合チェックが終了するまでの間、保持される。また、この仮リストは、上記メモリ 13 の所定領域、或いは CPU 12 内の一時記憶領域等に形成される。

【0093】

アプリケーション管理ブロック 64 は、アプリケーションリスト 63 に登録されている全てのアプリケーションに対して上記競合チェックが終了すると、上記



仮リストに登録されている各アプリケーションの遷移させるべき状態をそれぞれ検出する。そして、前述のように「起動不可」>「割込側待機」>「動作側待機」>「動作側終了」となっている「状態の優先度」に基づいて、競合する各アプリケーションの遷移させるべき状態を決定し、この結果を上記アプリケーションリスト63に反映する。

【0094】

この結果、図8に示すように実行中の各アプリケーションの「アプリケーション識別番号（アプリケーションID）」、「アプリケーション名」、「タスク識別番号（タスクID）」、「状態」等からなる実行リスト61、及び図9に示すように実行待ちの各アプリケーションの「アプリケーション識別番号（アプリケーションID）」、「アプリケーション名」、「タスク識別番号（タスクID）」、「状態」等からなる実行待ちリスト62が、アプリケーションリスト63内に形成されることとなる。

【0095】

なお、この図8及び図9に示す「状態」としては、上記決定されたそのアプリケーションの「遷移させるべき状態」を示す情報が登録され、例えば電話アプリケーションの場合は、上記接続状態、通話状態、保留状態・・・等に対応する情報が登録される。

【0096】

また、実行リスト61及び実行待ちリスト62に登録されるアプリケーションの数は、一つとは限らず、複数のアプリケーションが並行して実行され或いは実行待ちとなっている場合は、この複数のアプリケーションに対応する上記「アプリケーションID」や「アプリケーション名」等が登録される。

【0097】

図6のフローチャートのステップS1、ステップS2及びステップS4～ステップS9の流れは、アプリケーションリスト63に登録されている全てのアプリケーションに対する競合チェックが終了するまでの間、競合チェックの結果を仮リストに登録する動作を繰り返し実行し、アプリケーションリスト63に登録されている全てのアプリケーションに対する競合チェックが終了した際に、この仮

リストの登録内容をアプリケーションリスト63に反映する動作の流れを示している。

【0098】

次に、このようにしてアプリケーションリスト63に上記仮リストの結果が反映されると、アプリケーション制御ブロック65は、ステップS10において、アプリケーションリスト63の実行リスト61に登録されたアプリケーションを実行状態に遷移させると共に、実行待ちリスト62に登録されたアプリケーションを実行待ち状態に遷移させる。

【0099】

この後、アプリケーションマネージャ25は、「アプリケーションの起動要求」、「実行中のアプリケーションの終了」、「実行中のアプリケーションの状態の遷移」の検出待ちの状態となる。そして、これらいずれかが検出されたタイミングで、再度、この図6に示すフローチャートを実行することとなる。

【0100】

(電話アプリケーションが実行中の場合の競合管理)

このような動作を、図7の例を用いてさらに具体的に説明する。前述のように、この図7は、現在実行中のアプリケーションとして電話アプリケーションが実行されている場合に参照される競合条件テーブルの一例である。

【0101】

例えば、現在実行中の電話アプリケーションの状態が接続中(第1の状態)であるにもかかわらず、さらに電話アプリケーションの起動要求がなされた場合、競合条件テーブル26を参照すると、電話アプリケーションを重ねて起動しないことが好ましいことから、図7に示すように「起動不可」となっている。

【0102】

このため、アプリケーション制御ブロック65は、この場合、起動要求のあった電話アプリケーションの該要求をキャンセルし、現在実行中の電話アプリケーションを継続して実行制御する。

【0103】

これに対して、現在実行中の電話アプリケーションが接続中の状態(第1の状



態)で、電子メール(Eメール)の作成要求(或いは送信要求)がなされた場合、競合条件テーブル26を参照すると「割込側待機」となっている。このため、アプリケーション管理ブロック64は、起動要求のなされたEメールのアプリケーション(この場合、Eメールのアプリケーションは、割り込み側のアプリケーションとなる。)を上記実行待ちリスト62に登録して実行待ち状態とする。

【0104】

同様に、現在実行中の電話アプリケーションが通話中の状態(第2の状態)で、動画プレーヤの起動要求がなされた場合、競合条件テーブル26を参照すると「起動可能」となっている。このため、アプリケーション管理ブロック64は、動画プレーヤのアプリケーションを上記実行リスト61に登録する。これにより、アプリケーション制御ブロック65は、上記現在実行中の電話アプリケーション、及び上記動画プレーヤのアプリケーションを並行して実行制御することとなる。

【0105】


同様に、現在実行中の電話アプリケーションが接続中の状態(第1の状態)で、アドレス帳の起動要求がなされた場合、競合条件テーブル26を参照すると、「動作側待機」となっている。このため、アプリケーション管理ブロック64は、実行リスト61に登録されている現在実行中の電話アプリケーションを実行待ちリスト62に登録し直すと共に、代わって実行リスト61にアドレス帳のアプリケーションを新たに登録する。

【0106】

これにより、アプリケーション制御ブロック65は、それまで実行していた電話アプリケーションを実行待ち状態に遷移させ、起動要求のなされたアドレス帳を起動して実行制御することとなる。

【0107】

なお、この具体例の説明では、アプリケーション管理ブロック64は、競合条件テーブル26を用いた検出結果に基づいて、アプリケーションリスト63の登録内容を直接的に書き換えるが如く説明を行った。勿論、直接的に書き換えを行ってもよいのであるが、この実施の形態の場合は、上記「仮リスト」に判定結果



を一旦記憶し、全てのアプリケーションの判定が終了したタイミングで、該仮リストの内容をアプリケーションリスト63に反映させることは、前述のとおりである。

【0108】

(アドレス帳及び電話が競合した場合の競合管理)

次に、図10及び図11のシーケンス図(タイムチャート)を用いて、このような競合管理処理をさらに具体的に説明する。

【0109】

まず、図10は、先にアドレス帳のアプリケーションが起動されており、後から電話アプリケーションの起動要求があった場合における競合管理処理の流れを示している。

【0110】

なお、このアドレス帳のアプリケーションは、メモリ13の所定領域に設けられたアドレス帳の記憶領域に対して、通話先の携帯電話番号や携帯メールアドレス等の登録や削除等の編集を行うアプリケーションである。

【0111】

この図10において、ステップS11でアプリケーションランチャーからアプリケーションマネージャ25に対して電話アプリケーションの起動要求があった場合、アプリケーションマネージャ25のアプリケーション管理ブロック64は、ステップS12において、図7の例を用いて説明した競合条件テーブル26を参照し、前述の「競合チェック」を行う。

【0112】

この競合チェックにより、アドレス帳のアプリケーションを実行待ち状態とし、電話アプリケーションを起動させる競合チェックの結果(=動作側待機)が得られた場合、アプリケーション管理ブロック64は、アプリケーション制御ブロック65に対して、現在実行中のアドレス帳のアプリケーションの「待機要求」を送信する。

【0113】

この「待機要求」を受信すると、アプリケーション制御ブロック65は、ステ

ップS13に示すように、現在実行制御しているアドレス帳のアプリケーションに待機要求を送信することで、該アドレス帳のアプリケーションを待機状態に制御する。

【0114】

アドレス帳のアプリケーションは待機状態となると、ステップS14に示すように「待機完了」のデータを、アプリケーション制御ブロック65を介してアプリケーション管理ブロック64に送信する。

【0115】

次に、アプリケーション管理ブロック64は、この「待機完了」のデータを受信すると、アプリケーション制御ブロック65に対して電話アプリケーションの起動要求を送信する。この起動要求を受信すると、アプリケーション制御ブロック65は、ステップS15に示すように電話アプリケーションに対して起動要求を送信し、該電話アプリケーションを起動制御する。

【0116】

なお、起動が完了すると、ステップS16に示すように電話アプリケーション側からアプリケーション管理ブロック64に対して「起動完了通知」がなされる。

【0117】

次に、電話アプリケーションの実行が進行し、通話の終了等により電話アプリケーションの実行が終了すると、ステップS17に示すように電話アプリケーション側からアプリケーション管理ブロック64に対して「終了通知」がなされる。

【0118】

次に、この終了通知を受信すると、アプリケーション管理ブロック64は、ステップS18において、アプリケーションリスト63に登録されている実行中のアプリケーション及び実行待ちのアプリケーションを参照し、前述の競合チェックを行う。そして、この競合チェックにより、先程、待機状態としたアドレス帳のアプリケーションを起動させてもよい場合に、アプリケーション制御ブロック65に対してアドレス帳のアプリケーションの起動要求を送信する。



【0119】

アプリケーション制御ブロック65は、この起動要求を受信すると、ステップS19に示すように先程、待機状態としたアドレス帳のアプリケーションを再実行制御する。この再実行制御によりアドレス帳のアプリケーションが再び実行状態に遷移すると、このアドレス帳のアプリケーションからアプリケーション管理ブロック64に対して「起動完了通知」がなされる。

【0120】

これにより、アドレス帳の編集作業等に割り込むかたちで電話アプリケーションが実行され、終話後に、再度、アドレス帳のアプリケーションが実行され、アドレス帳の編集作業等が行われることとなる。

【0121】

(Eメール及び電話が競合した場合の競合管理)

次に、図11は、先に電話アプリケーションが起動されており、後からEメールのアプリケーションの起動要求があった場合における競合管理処理の流れを示している。

【0122】

なお、このEメールのアプリケーションは、携帯メールの送受信、送受信した携帯メールの削除等の編集や携帯メールの作成等を行うアプリケーションである。

【0123】

この図11において、ステップS21でアプリケーションランチャーからアプリケーションマネージャ25に対してEメールのアプリケーションの起動要求があった場合、アプリケーションマネージャ25のアプリケーション管理ブロック64は、ステップS22において、上記競合条件テーブル26を参照し、前述の「競合チェック」を行う。

【0124】

この競合チェックにより、起動要求のなされたEメールのアプリケーションを実行待ち状態とし、電話アプリケーションを継続して実行する競合チェックの結果(=割込側待機)が得られた場合、アプリケーション管理ブロック64は、ア



アプリケーション制御ブロック 65 に対して、起動要求のなされた E メールアプリケーションの「待機要求」を送信する。

【0125】

アプリケーション制御ブロック 65 は、この「待機要求」を受信すると、ステップ S23 において、起動要求のなされた E メールアプリケーションを待機状態に制御すると共に、現在実行制御している電話アプリケーションを継続して実行制御する。

【0126】

次に、電話アプリケーションの実行が進行し、通話の終了等により電話アプリケーションの実行が終了すると、ステップ S24 に示すように電話アプリケーション側からアプリケーション管理ブロック 64 に対して「終了通知」がなされる。

【0127】

この終了通知を受信すると、アプリケーション管理ブロック 64 は、ステップ S25 において、アプリケーションリスト 63 に登録されている実行中のアプリケーション及び実行待ちのアプリケーションを参照し、前述の競合チェックを行う。そして、この競合チェックにより、先程、待機状態とした E メールアプリケーションを起動させてもよい場合に、アプリケーション制御ブロック 65 に対して E メールアプリケーションの起動要求を送信する。

【0128】

アプリケーション制御ブロック 65 は、この起動要求を受信すると、ステップ S26 に示すように先程、待機状態とした E メールアプリケーションを起動制御する。これにより、終話後に、Eメールの送受信や作成等が行われることとなる。

【0129】

なお、ステップ S27 に示すように、Eメールアプリケーションの起動が完了すると、該 Eメールアプリケーションからアプリケーション管理ブロック 64 に対して「起動完了通知」がなされる。

【0130】



[実施の形態の効果]

以上の説明から明らかなように、当該実施の形態の携帯電話機は、アプリケーションマネージャ25が、現在実行中のアプリケーションを、アプリケーションリスト63の実行リスト61に登録し、実行待ちとなっているアプリケーションを実行待ちリスト62に登録する。

【0131】

そして、「アプリケーションの起動要求を受け付けた際」、「実行中のアプリケーションの終了を検出した際」、「実行中のアプリケーションの状態遷移を検出した際」等に、実行リスト61及び実行待ちリスト62を参照して、実行するアプリケーション及び実行待ちさせるアプリケーションを決定して、マルチタスク処理における各アプリケーションの競合管理を行う。これにより、以下の効果を得ることができる。

【0132】

実行中のアプリケーションの現在の状態の他、実行待ちとなっているアプリケーションの状態を参照し、実行中のアプリケーション及び実行待ちのアプリケーションを総合的に判断して、実行するアプリケーション及び実行待ちとするアプリケーションを決定するようになっている。

【0133】


このため、実行待ちとなっているアプリケーションの実行の再開を保証することができる。

【0134】

また、従来のアプリケーションの競合管理は、例えば表計算ソフトウェアの実行中にユーザから文書作成ソフトウェアの実行指示があった場合、この実行指示を優先して文書作成ソフトウェアの立ち上げを行うようになっている。

【0135】

しかし、当該実施の形態の携帯電話機の場合、実行中のアプリケーション及び実行待ちのアプリケーションを総合的に判断して、実行するアプリケーション及び実行待ちとするアプリケーションを決定するようになっているため、ユーザから実行指示がなされた場合でも、表計算ソフトウェアが所定の実行状態の場合は



、文書作成ソフトウェアが起動しないようにすることができる。従って、より細かなアプリケーションの競合管理を可能とすることができる。

【0136】

また、実行するアプリケーション及び実行待ちとするアプリケーションは、アプリケーションマネージャ25が、アプリケーションリスト63及び競合条件テーブル26を参照して決定するようになっている。すなわち、そのアプリケーションの実行の是非は、アプリケーションマネージャ25が判断するようになっている。

【0137】

このため、アプリケーションをプログラミングする際に、他のアプリケーションとの競合関係を考慮することなくプログラミングを行うことを可能とすることができる。

【0138】

さらに、このように他のアプリケーションとの競合関係を考慮することなくプログラミングを行うことを可能とすることができるため、あるアプリケーションに仕様変更が生じた場合は、そのアプリケーションに対してのみ、仕様変更を加えればよいこととなる。

【0139】

このため、あるアプリケーションに仕様変更が生じた場合に、他の全てのアプリケーションに対しても、この仕様変更に対応した仕様変更を加え、多数のアプリケーションの再プログラミング作業や再インストール作業（アプリケーションの設計及び実装の変更作業）等の大変面倒な作業が必要となる不都合を防止することができる。

【0140】

そして、当該実施の形態の携帯電話機は、このようにアプリケーションのプログラミングを行う際に、他のアプリケーションの競合関係を考慮しなくてもよいため、ソフトウェアの開発を大幅に簡単化することができる。

【0141】

〔他の適用分野〕

上述の実施の形態の説明では、本発明を、携帯電話機に適用することとしたが、本発明はこの他、PHS電話機（PHS:Personal Handyphone System）やPDA装置（PDA:Personal Digital(Data) Assistant）等の移動体通信端末装置の他、パーソナルコンピュータ装置等の端末装置に適用してもよい。

【0142】

また、上述の実施の形態の説明では、アプリケーションの競合管理を行うようにしたが、これは、タスク毎の競合管理を行うようにしてもよい。

【0143】

最後に、上述の実施の形態は本発明の一例である。このため、本発明は、上述の実施の形態に限定されることはなく、該実施の形態以外であっても、本発明に係る技術的思想を逸脱しない範囲であれば、設計等に応じて種々の変更が可能であることは勿論であることを付け加えておく。

【0144】

【発明の効果】

本発明は、実行待ちとなっているタスクの実行の再開を保証することができる。

【0145】

また、タスクのプログラミングを行う際に、他のタスクとの競合関係を考慮することなくプログラミングを行うことを可能とすることができる。

【0146】

また、他のタスクとの競合関係を考慮することなくプログラミングが可能となるため、あるタスクに仕様変更が生じた場合は、そのタスクに対してのみ、仕様変更を加えればよいこととなる。

【0147】

このため、あるタスクに仕様変更が生じた場合に、他の全てのタスクに対して、この仕様変更に対応した仕様変更を加える必要が生じ、多数のタスクの再プログラミング作業や再インストール作業（タスクの設計及び実装の変更作業）等の大変面倒な作業が発生する不都合を防止することができる。

【0148】

そして、このようにタスクのプログラミングを行う際に、他のタスクの競合関係を考慮しなくてもよいため、ソフトウェアの開発を大幅に簡単化することができる。

【図面の簡単な説明】

【図 1】

本発明を適用した実施の形態となる携帯電話機の全体的なブロック図である。

【図 2】

実施の形態の携帯電話機の主要部の機能ブロック図である。

【図 3】

実施の形態の携帯電話機で実行される各アプリケーションの機能ブロック図である。

【図 4】

実施の形態の携帯電話機に設けられているアプリケーションマネージャの機能ブロック図である。

【図 5】

各アプリケーションの状態遷移を説明するための模式図である。

【図 6】

実施の形態の携帯電話機に設けられているアプリケーションマネージャのアプリケーションの競合管理動作を説明するためのフローチャートである。

【図 7】

実施の形態の携帯電話機に設けられている競合条件テーブルの一例を示す図である。

【図 8】

アプリケーションマネージャのアプリケーションリストに設けられている実行リストを示す図である。

【図 9】

アプリケーションマネージャのアプリケーションリストに設けられている実行待ちリストを示す図である。

【図 10】



アドレス帳のアプリケーションの実行中に、電話アプリケーションの実行要求があった場合におけるアプリケーションマネージャの競合管理動作を説明するためのタイムチャートである。

【図 11】

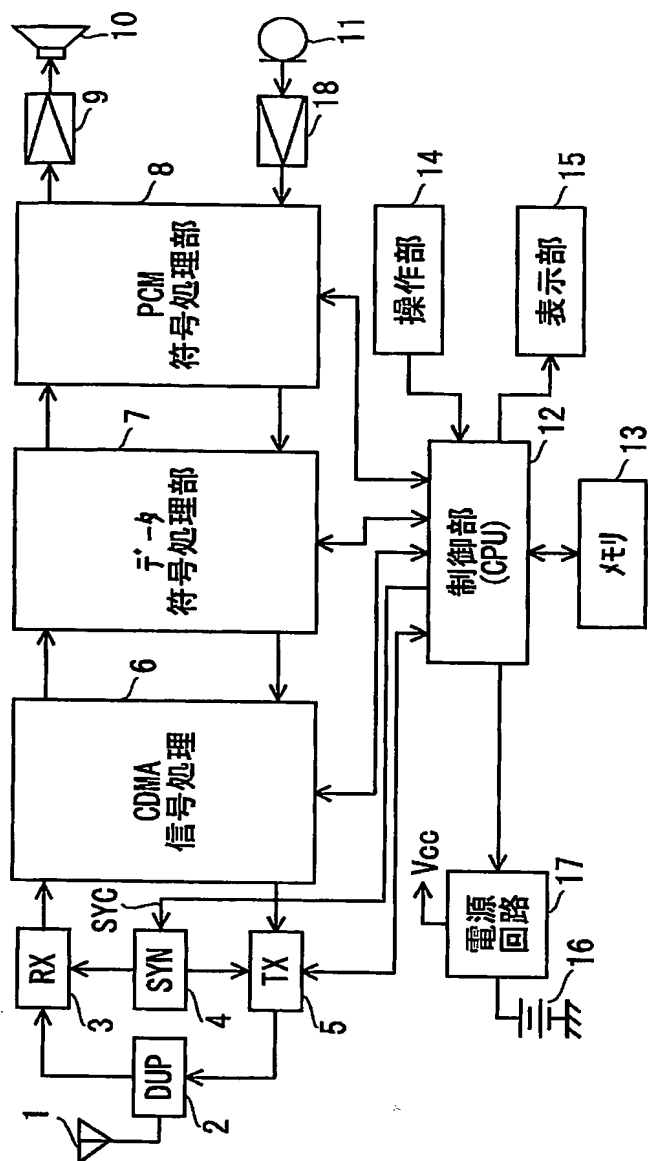
電話アプリケーションの実行中に、Eメールのアプリケーションの実行要求があった場合におけるアプリケーションマネージャの競合管理動作を説明するためのタイムチャートである。

【符号の説明】

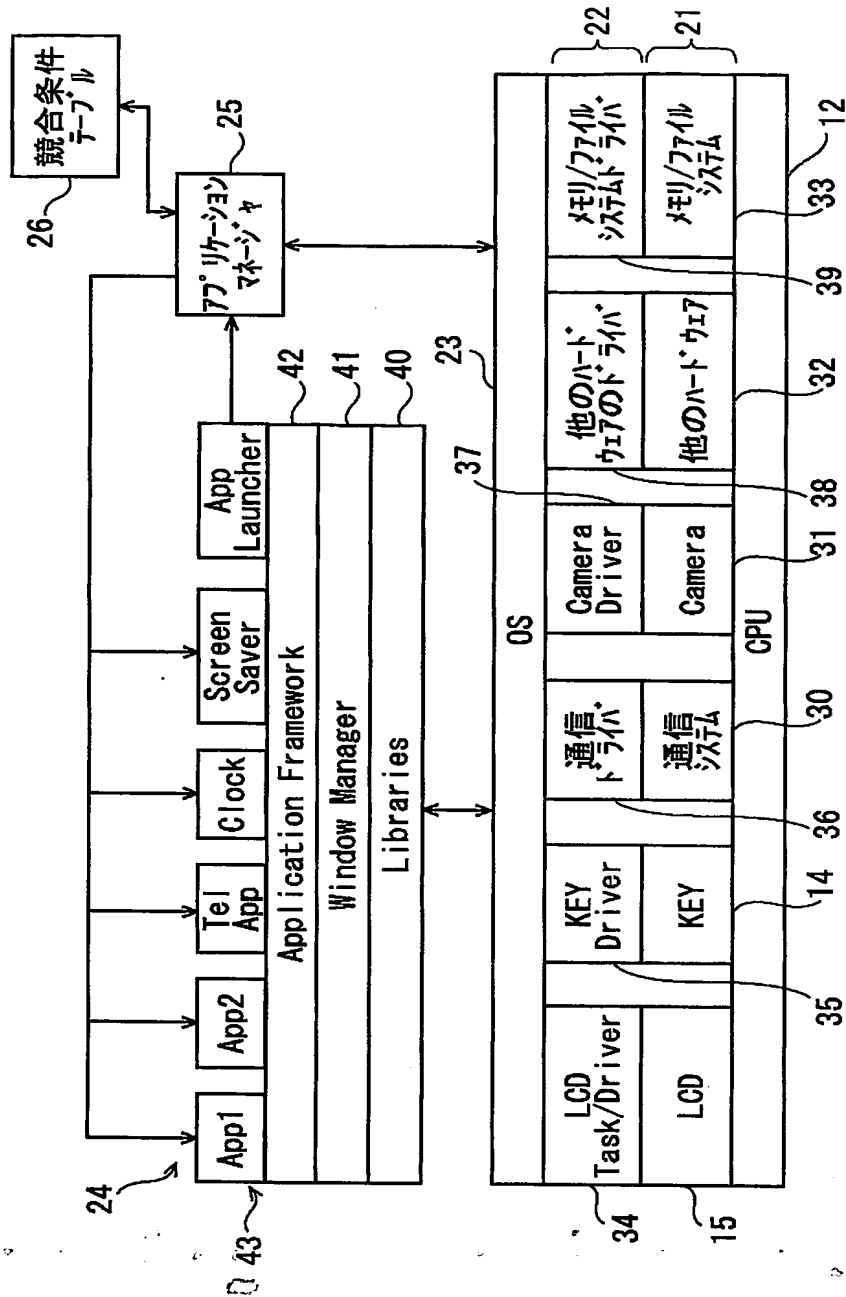
12…制御部（CPU）、21…ハードウェア群、22…ドライバ群、23…オペレーションシステム（OS）、24…アプリケーション群、25…アプリケーションマネージャ、26…競合条件テーブル、60…コミュニケーションブロック、61…実行リスト、62…実行待ちリスト、63…アプリケーションリスト、64…アプリケーション管理ブロック、65…アプリケーション制御ブロック

【書類名】 図面

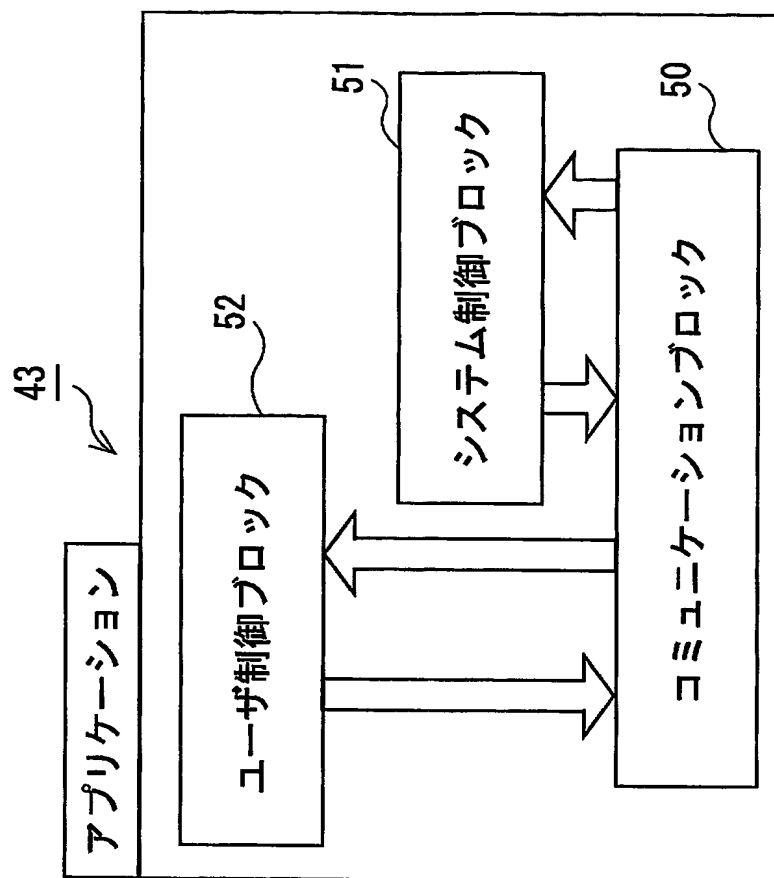
【図 1】



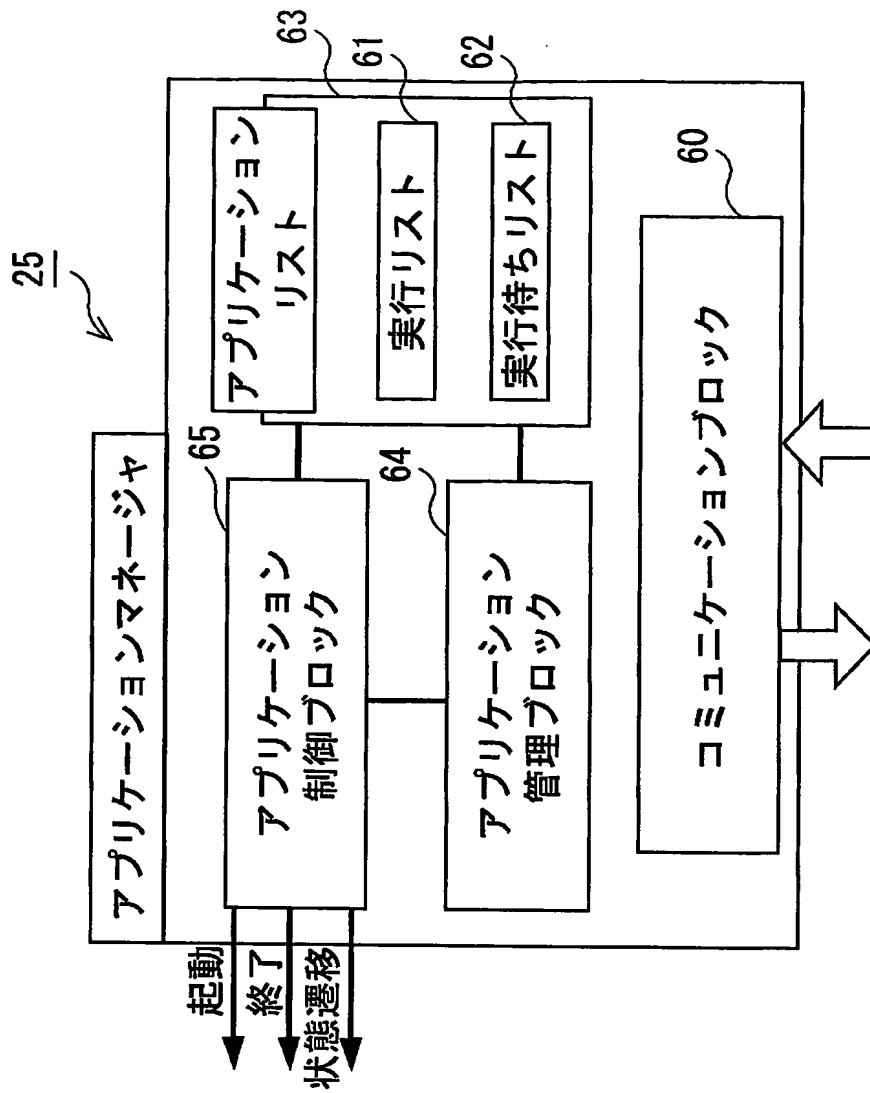
【図 2】



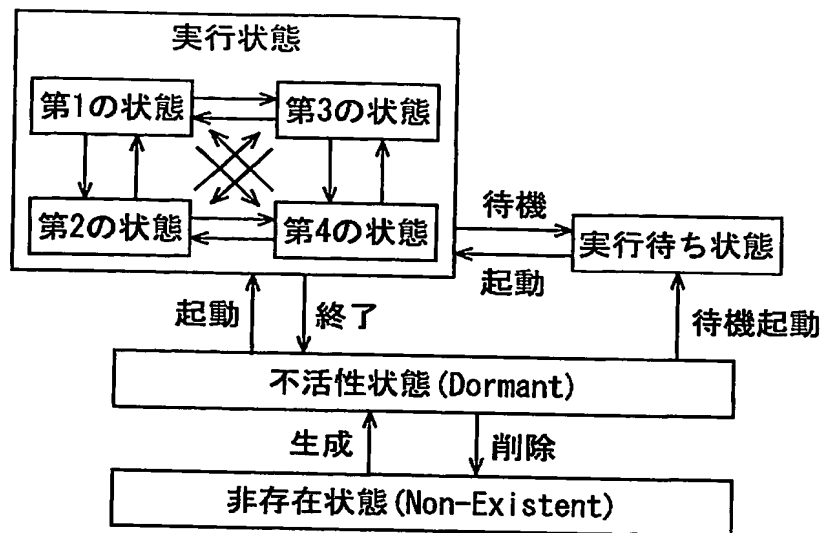
【図 3】



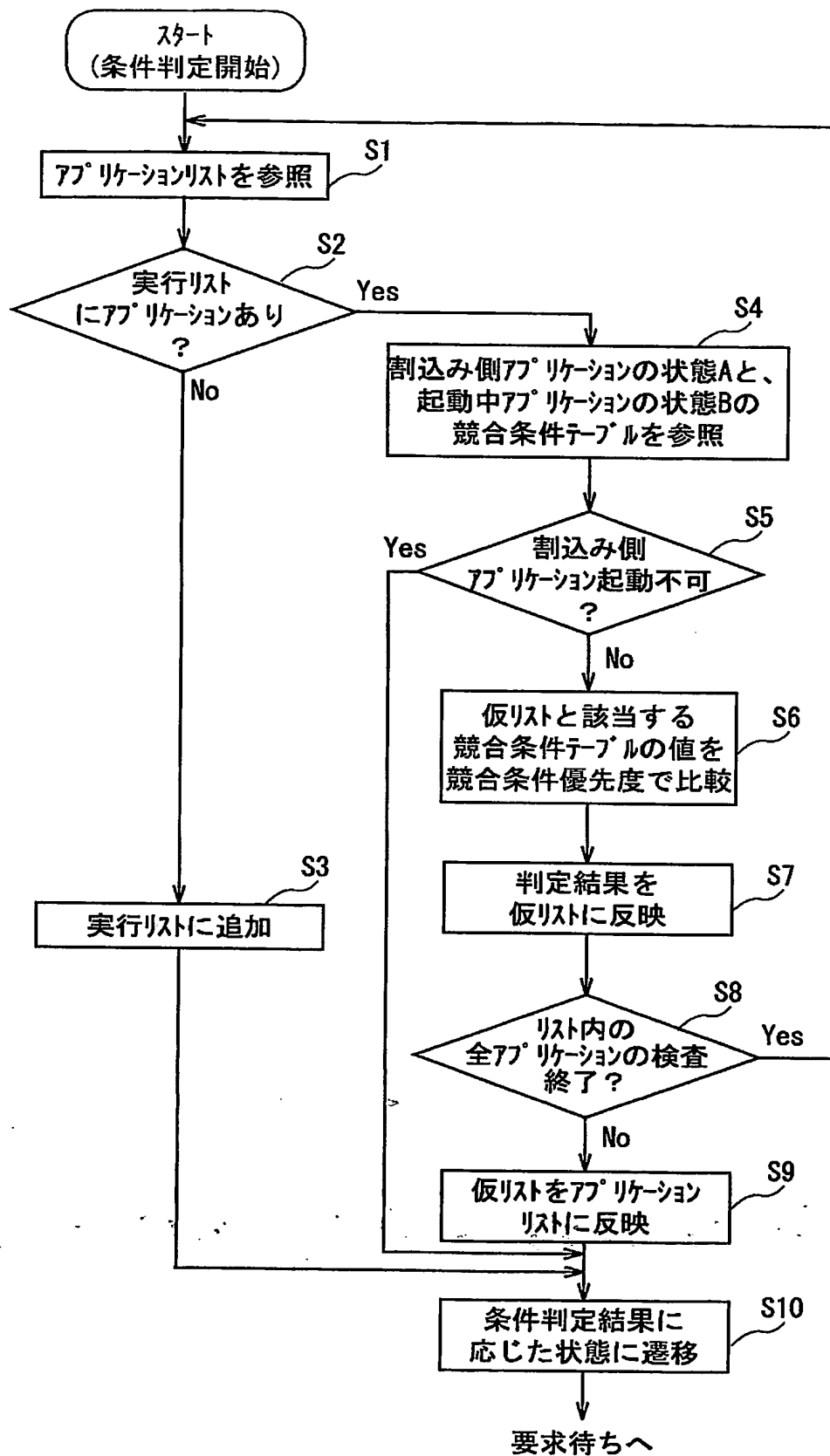
【図 4】



【図 5】



【図 6】



【図 7】

26

状態 アプリケーション	第1の状態	第2の状態	第3の状態	第4の状態	第Nの状態
電話	起動不可	起動不可	起動不可	起動不可	起動不可
待ち受け	動作側待機	動作側待機	動作側終了	動作側終了	動作側終了
アドレス帳	動作側待機	動作側待機	動作側待機	動作側待機	動作側終了
Eメール	割込側待機	起動可能	動作側待機	動作側待機	動作側終了
ブラウザ	起動可能	起動可能	動作側待機	起動可能	動作側終了
動画プレーヤ	起動可能	起動可能	動作側待機	動作側待機	起動可能
メモ帳	起動可能	起動可能	動作側待機	動作側終了	割込側待機

【図 8】

61

実行リスト

アプリケーションID	アプリケーションID	...	アプリケーションID
アプリケーション名	アプリケーション名		アプリケーション名
タスクID	タスクID		タスクID
状態	状態		状態

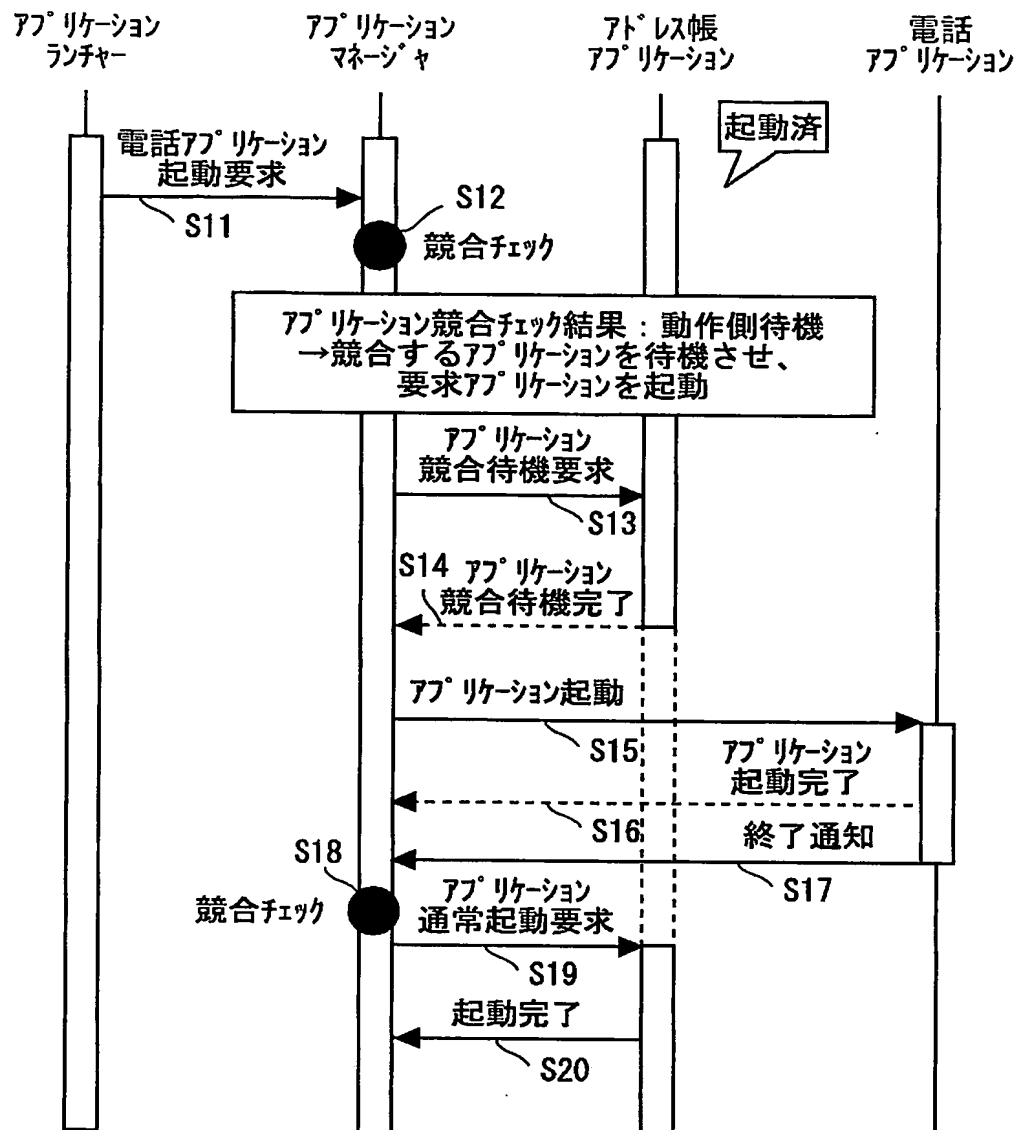
【図 9】

62

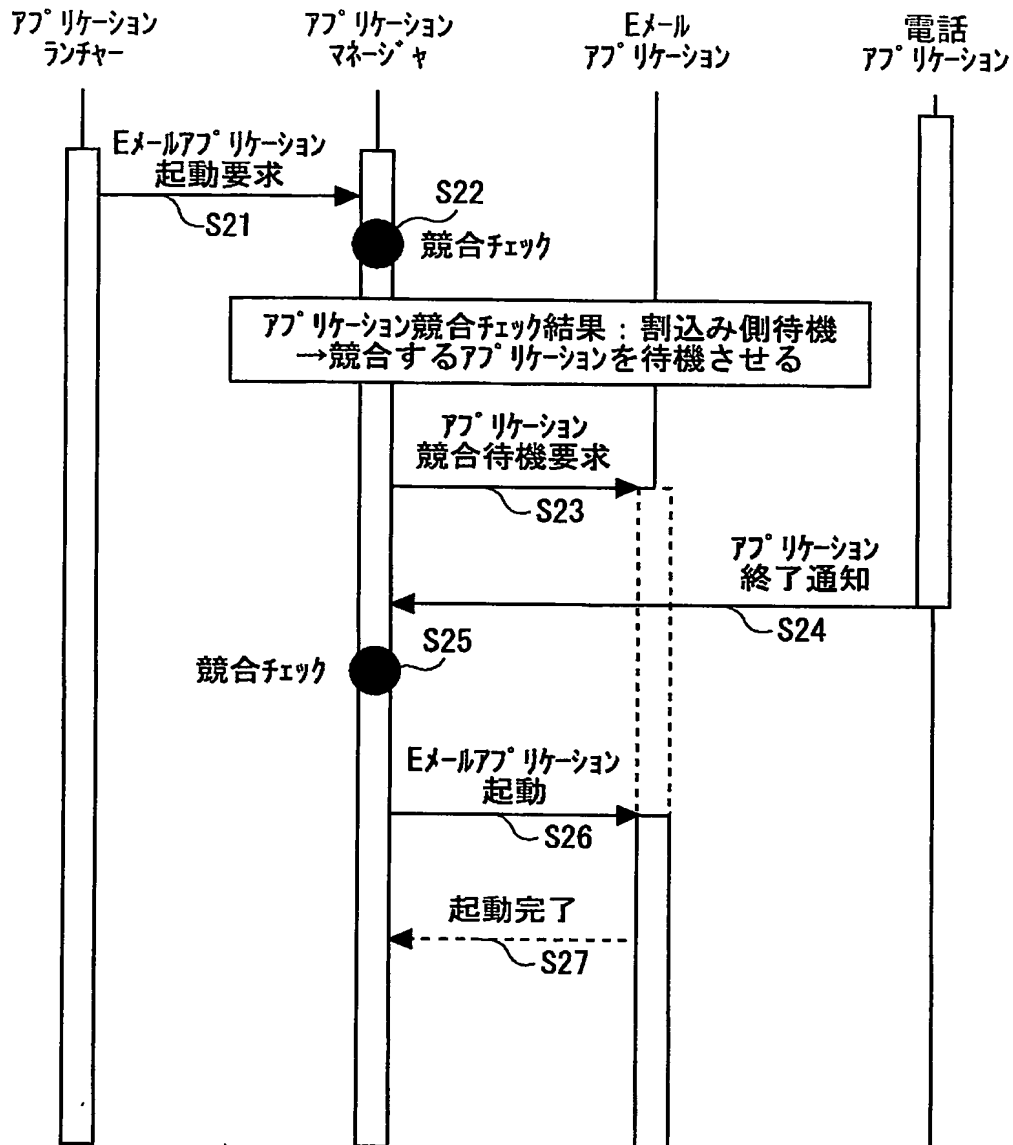
実行待ちリスト

アプリケーションID	アプリケーションID	...	アプリケーションID
アプリケーション名	アプリケーション名		アプリケーション名
タスクID	タスクID		タスクID
状態	状態		状態

【図10】



【図 11】





【書類名】 要約書

【要約】

【課題】 マルチタスク処理における新たなアプリケーションの競合管理方法を提供する。

【解決手段】 アプリケーションマネージャ25に、現在実行中のアプリケーションを登録する実行リスト61及び待機中のアプリケーションを登録する実行待ちリスト62を設ける。アプリケーション管理ブロック64は、アプリケーションの競合が生じた場合、上記各リスト61、62に登録されている実行中のアプリケーションの状態及び待機中のアプリケーションの状態に対応する競合条件テーブルを参照し、各状態の優先度に基づいて、各アプリケーションの遷移させるべき状態（起動、終了、実行待ち等）を決定する。そして、アプリケーション制御ブロック65が、このアプリケーション管理ブロック64により決定された状態に、各アプリケーションを遷移させることで、各アプリケーションの競合管理を行う。

【選択図】 図4

認定・付加情報

特許出願の番号	特願 2003-165742
受付番号	50300971953
書類名	特許願
担当官	第七担当上席 0096
作成日	平成15年 6月11日

<認定情報・付加情報>

【提出日】 平成15年 6月10日

次頁無

特願 2003-165742

出 願 人 履 歴 情 報

識別番号

[501431073]

1. 変更年月日

2001年11月 6日

[変更理由]

新規登録

住 所

東京都港区港南1丁目8番15号 Wビル

氏 名

ソニー・エリクソン・モバイルコミュニケーションズ株式会社